NAME

crypttab – static information about encrypted filesystems

DESCRIPTION

The file /etc/crypttab contains descriptive information about encrypted filesystems. crypttab is only read by programs (e.g. **cryptdisks_start** and **cryptdisks_stop**), and not written; it is the duty of the system administrator to properly create and maintain this file. Each filesystem is described on a separate line; fields on each line are separated by tabs or spaces. Lines starting with “#” are comments, empty lines are ignored. The order of records in crypttab is important because the init scripts sequentially iterate through crypttab doing their thing.

The first field, *target*, describes the mapped device name. It must be a plain filename without any directory components. A mapped device which encrypts/decrypts data to/from the *source device* will be created at /dev/mapper/target by **cryptsetup**.

The second field, *source device*, describes either the block special device or file that contains the encrypted data. Instead of giving the *source device* explicitly, the UUID (resp. LABEL, PARTUUID and PARTLABEL) is supported as well, using “UUID=<uuid>” (resp. “LABEL=<label>”, “PARTUUID=<partuuid>” and “PARTLABEL=<partlabel>”).

The third field, *key file*, describes the file to use as a key for decrypting the data of the *source device*. In case of a *keyscript*, the value of this field is given as argument to the keyscript. Values with spaces and special characters need to be escaped using octal sequences, like for **fstab**(5). Note that the *entire* key file will be used as the passphrase; the passphrase must *not* be followed by a newline character.

It can also be a device name (e.g. /dev/urandom), note however that LUKS requires a persistent key and therefore does *not* support random data keys.

If the *key file* is the string *none*, a passphrase will be read interactively from the console. In this case, the options check, checkargs and tries may be useful.

The fourth field, *options*, describes the cryptsetup options associated with the encryption process. At minimum, the field should contain either the string *luks* respectively *tcrypt* or the *cipher*, *hash* and *size* options. Some options can be changed on active devices using **cryptsetup refresh [<options>] <name>**. Moreover some options can be permanently written to the metada of LUKS2 headers using the **−−persistent** option flag.

Options are in the format: *key*=*value* [,*key*=*value* ...]. The supported options are described below.

Note that all four fields are mandatory and that a missing field will lead to unspecified behaviour.

ON DIFFERENT CRYPTTAB FORMATS

Please note that there are several independent cryptsetup wrappers with their own *crypttab* format. This manpage covers Debian's implementation for *initramfs* scripts and *SysVinit* init scripts. *systemd* brings its own *crypttab* implementation. We try to cover the differences between the *systemd* and our implementation in this manpage, but if in doubt, better check the *systemd* **crypttab**(5) manpage, e.g. online at **https://www.freedesktop.org/software/systemd/man/crypttab.html**.

OPTIONS

cipher=<cipher>
: Encryption algorithm (ignored for LUKS and TCRYPT devices). See **cryptsetup −c**.

size=<size>
: Encryption key size (ignored for LUKS and TCRYPT devices). See **cryptsetup −s**.

sector−size=<bytes>
: Sector size. See **cryptsetup**(8) for possible values and the default value of this option.

hash=<hash>
Hash algorithm (ignored for LUKS and TCRYPT devices). See **cryptsetup −h**.

offset=<offset>
Start offset (ignored for LUKS and TCRYPT devices). Uses **cryptsetup −o**.

skip=<skip>
Skip sectors at the beginning (ignored for LUKS and TCRYPT devices). Uses **cryptsetup −p**.

keyfile−offset=<keyfile−offset>
Specifies the number of bytes to skip at the start of the key file.

keyfile−size=<keyfile−size>
Specifies the maximum number of bytes to read from the key file. The default is to read the whole file up to the compiled−in maximum, that can be queried with **cryptsetup −−help**. This option is ignored for plain dm−crypt devices, as the key file size is then given by the encryption key size (option *size*).

keyslot=<slot>, *key−slot*=<slot>
Key slot (ignored for non−LUKS devices). See **cryptsetup −S**.

header=<path>
Detached header file (ignored for plain dm−crypt devices). See **cryptsetup −−header**.

verify
Verify password. Uses **cryptsetup −y**.

readonly, *read−only*
Set up a read−only mapping.

tries=<num>
Try to unlock the device <num> before failing. It's particularly useful when using a passphrase or a *keyscript* that asks for interactive input. If you want to disable retries, pass “tries=1”. Default is “3”. Setting “tries=0” means infinitive retries.

discard
Allow using of discards (TRIM) requests for device.

Starting with Debian 10 (Buster), this option is added per default to new dm−crypt devices by the Debian Installer. If you don't care about leaking access patterns (filesystem type, used space) and don't have hidden truecrypt volumes inside this volume, then it should be safe to enable this option. See the following warning for further information.

WARNING: Assess the specific security risks carefully before enabling this option. For example, allowing discards on encrypted devices may lead to the leak of information about the ciphertext device (filesystem type, used space etc.) if the discarded blocks can be located easily on the device later.

luks
Force LUKS mode. When this mode is used, the following options are ignored since they are provided by the LUKS header on the device: *cipher=*, *hash=*, *size=*

plain
Force plain encryption mode.

tcrypt
Use TrueCrypt encryption mode. When this mode is used, the following options are ignored since they are provided by the TrueCrypt header on the device or do not apply: *cipher=*, *hash=*, *keyfile−offset=*, *keyfile−size=*, *size=*

veracrypt, *tcrypt−veracrypt*
Use VeraCrypt extension to TrueCrypt device. Only useful in conjunction with *tcrypt* option (ignored for non−TrueCrypt devices).

tcrypthidden, *tcrypt−hidden*

Use hidden TCRYPT header (ignored for non−TCRYPT devices).

swap

Run **mkswap** on the created device.

This option is ignored for *initramfs* devices.

tmp=<tmpfs>

Run **mkfs** with filesystem type <tmpfs> on the created device. Default is ext4.

This option is ignored for *initramfs* devices.

check=<check>

Check the content of the target device by a suitable program; if the check fails, the device is removed. If a program is provided as an argument, it is run, giving the decrypted volume (target device) as first argument, and the value of the checkargs option as second argument. Cryptdisks/cryptroot searches for the given program in /lib/cryptsetup/checks/ first, but full path to program is supported as well.

Default is set in /etc/default/cryptdisks (blkid).

This option is specific to the Debian *crypttab* format. It's not supported by *systemd*.

checkargs=<arguments>

Give <arguments> as the second argument to the check script. See the CHECKSCRIPTS section for more information.

This option is specific to the Debian *crypttab* format. It's not supported by *systemd*.

initramfs

The initramfs hook processes the root device, any resume devices and any devices with the *initramfs* option set. These devices are processed within the initramfs stage of boot. As an example, that allows the use of remote unlocking using dropbear.

This option is specific to the Debian *crypttab* format. It's not supported by *systemd*.

noearly

The cryptsetup init scripts are invoked twice during the boot process − once before lvm, raid, etc. are started and once again after that. Sometimes you need to start your encrypted disks in a special order. With this option the device is ignored during the first invocation of the cryptsetup init scripts.

This option is ignored for *initramfs* devices and specific to the Debian *crypttab* format. It's not supported by *systemd*.

noauto

Entirely ignore the device at the boot process. It's still possible to map the device manually using cryptdisks_start.

This option is ignored for *initramfs* devices and specific to the Debian *crypttab* format. It's not supported by *systemd*.

loud

Be loud. Print warnings if a device does not exist. This option overwrites the option *loud*.

This option is ignored for *initramfs* devices and specific to the Debian *crypttab* format. It's not supported by *systemd*.

quiet

Be quiet. Don't print warnings if a device does not exist. This option overwrites the option *loud*.

This option is ignored for *initramfs* devices and specific to the Debian *crypttab* format. It's not supported by *systemd*.

keyscript=<path>

The executable at the indicated path is executed with the value of the *third field* as only argument. The keyscript output is passed to cryptsetup as decyption key. When used in initramfs, the executable either needs to be self−contained (i.e. does'nt rely on any external program which is not present in the initramfs environment) or the dependencies have to added to the initramfs image by other means.

LIMITATIONS: All binaries and files on which the keyscript depends must be available at the time of execution. Special care needs to be taken for encrypted filesystems like /usr or /var. As an example, unlocking encrypted /usr must not depend on binaries from /usr/(s)bin.

This option is specific to the Debian *crypttab* format. It's not supported by *systemd*.

WARNING: With systemd as init system, this option might be ignored. At the time this is written (December 2016), the systemd cryptsetup helper doesn't support the keyscript option to /etc/crypttab. For the time being, the only option to use keyscripts along with systemd is to force processing of the corresponding crypto devices in the initramfs. See the 'initramfs' option for further information.

All fields of the appropriate crypttab entry are available to the keyscript as exported environment variables:

CRYPTTAB_NAME

The target name

CRYPTTAB_SOURCE

The source device

CRYPTTAB_KEY

The key file

CRYPTTAB_OPTIONS

A list of exported crypttab options

CRYPTTAB_OPTION_<option>

The value of the appropriate crypttab option, with value set to 'yes' in case the option is merely a flag. For option aliases, such as 'readonly' and 'read−only', the variable name refers to the first alternative listed (thus 'CRYPTTAB_OPTION_readonly' in that case). If the crypttab option name contains '−' characters, then they are replaced with '_' in the exported variable name. For instance, the value of the 'CRYPTTAB_OPTION_keyfile_offset' environment variable is set to the value of the 'keyfile−offset' crypttab option.

CRYPTTAB_TRIED

Number of previous tries since start of cryptdisks (counts until maximum number of tries is reached).

CHECKSCRIPTS

blkid

Checks for any known filesystem. Supports a filesystem type as argument via <checkargs>:

- no checkargs − succeeds if any valid filesystem is found on the device.
- "none" − succeeds if no valid filesystem is found on the device.
- "ext4" [or another filesystem type like xfs, swap, crypto_LUKS, ...] − succeeds if ext4 filesystem is found on the device.

un_blkid

Checks for no known filesystem. Supports a filesystem type as argument via <checkargs>:

- no checkargs − succeeds if no valid filesystem is found on the device.
- "ext4" [or another filesystem type like xfs, swap, crypto_LUKS, ...] − succeeds if no ext4 filesystem is found on the device.

EXAMPLES

```
# Encrypted swap device
cswap /dev/sda6 /dev/urandom cipher=aes−xts−plain64,size=256,hash=sha1,swap

# Encrypted LUKS disk with interactive password, identified by its UUID, discard enabled
cdisk0 UUID=12345678−9abc−def012345−6789abcdef01 none luks,discard

# Encrypted TCRYPT disk with interactive password, discard enabled
tdisk0 /dev/sr0 none tcrypt,discard

# Encrypted ext4 disk with interactive password, discard enabled
# − retry 5 times if the check fails
cdisk1 /dev/sda2 none discard,cipher=aes−xts−plain64,size=256,hash=sha1,checkargs=ext4,tries=5

# Encrypted disk with interactive password, discard enabled
# − use a nondefault check script
# − no retries
cdisk2 /dev/sdc1 none discard,cipher=aes−xts−plain64,size=256,hash=sha1,check=customscript,tries=1

# Encrypted disk with interactive password, discard enabled
# − Twofish as the cipher, RIPEMD−160 as the hash
cdisk3 /dev/sda3 none dscard,cipher=twofish,size=256,hash=ripemd160
```

ENVIRONMENT

CRYPTDISKS_ENABLE

Set to *yes* to run cryptdisks initscripts at startup. Set to *no* to disable cryptdisks initscripts. Default is *yes*.

CRYPTDISKS_MOUNT

Specifies the mountpoints that are mounted before cryptdisks is invoked. Takes mountpoints configured in /etc/fstab as arguments. Separate mountpoints by space. This is useful for keys on removable devices, such as cdrom, usbstick, flashcard, etc. Default is unset.

CRYPTDISKS_CHECK

Specifies the default checkscript to be run against the target device, after cryptdisks has been invoked. The target device is passed as the first and only argument to the checkscript. Takes effect if the *check* option is given in crypttab with no value. See documentation for *check* option above for more information.

KNOWN UPGRADE ISSUES

The upstream defaults for encryption cipher, hash and keysize have changed several times in the past, and they're expected to change again in future, for example if security issues arise. On LUKS devices, the used settings are stored in the LUKS header, and thus don't need to be configured in /etc/crypttab. For plain dm−crypt devices, no information about used cipher, hash and keysize are available at all. Therefore we strongly suggest to configure the cipher, hash and keysize in /etc/crypttab for plain dm−crypt devices, even if they match the current default.

SEE ALSO

cryptsetup(8), **cryptdisks_start**(8), **cryptdisks_stop**(8),
/usr/share/doc/cryptsetup−initramfs/README.initramfs.gz

AUTHOR

This manual page was originally written by Bastian Kleineidam <calvin@debian.org> for the Debian distribution of cryptsetup. It has been further improved by Michael Gebetsroither <michael.geb@gmx.at>, David Härdeman <david@hardeman.nu> and Jonas Meurer <jonas@freesources.org>.